Speller with Word Completion and Prediction using Eye Tracker

Sudhakaran J. Jain (S3558487)
University of Groningen, Netherlands
s.j.jain@student.rug.nl

ABSTRACT

This paper describes an approach on creating a system that will serve as a effective medium for paralyzed and other physically challenged people to communicate with others by tracking their eye movements. This project aims to carry forward the previous work on a speller [7] that types letters based on user's eye movement using an eye tracker and further attempting to complete or predict the words that user intends to write. To build this system we make use of a word repository of most frequently occurring words along with a 5-gram word repository containing the most often used 5-gram sentences or phrases. Finally, to test our system, we conduct an experiment using seven random participants. We may then discuss and conclude on the results of data gathered from each participant.

Author Keywords

Eye tracking; pupil; blink; word; prediction; speller; sentence; python; pylink; opensesame; pygaze.

INTRODUCTION

There are several communication devices available out there designed specifically for paralyzed people. Such devices help the disabled to efficiently speak to others by doing minimum or negligible amount of physical work. One such device is the Speller which was designed to write alphabetical letters on a computer screen based on pupil dilation as input using an eye tracking device. Our aim is to further this previous work by adding word completion and prediction feature using intentional eye blinks to write a sentence in a much easier and faster way. From now on we may call the existing Speller as 'traditional Speller' and our new proposed system as 'upgraded Speller' in our future references to avoid confusion.

In traditional Speller, the user tries to choose an intended letter among all other letters and control symbols present on the screen by looking at the center but focusing on the intended letter. The average time taken to spell each letter in this setup was approximately known to be 51 seconds with an average accuracy around 70%. Moreover, after every word is completed, the user had to select 'space' symbol before beginning with another word in order to form a meaningful sentence. So, if the user wanted to write a bigger sentence, it became a tedious task.

We attempt to solve all these drawbacks in our upgraded Speller and offer a better communication medium for the paralyzed for their daily use. In order to test it, we conducted an experiment where participants were told to write a simple sentence using this system.

Results show that upgraded Speller has faster sentence completion. We were able to reliably distinguish between intentional eye blinks and normal ones. However, looking away from the screen is also classified as an intentional blink as the eye tracker loses track of the pupil. The addition of blink detection showed some improvements and/or faster word completion, but it was not as high as expected.

RELATED WORK

The traditional Speller was built based on the concept of pupillary response [1]. This is a normal phenomenon in which pupils of the eyes either widen or contract in response to the intensity of light reaching the retina, thereby adapting to different levels of lightness/darkness. Though, such responses can have other causes too.

The first screen of traditional Speller contains all the alphabetical letter along with some control symbols namely delete, stop and spaces. The letters are grouped in circles and these circles are arranged in circular format as shown in figure. These circles keep on changing colors from black to white and vice versa alternatively. The user is instructed to look at the center but still focus on the letter he/she wants to select. In such a scenario, the pupil size of user starts to change according to the changes in color of his/her intended letter. The Speller then detects these changes through eye tracker and tries to identify which circles the user might be focusing at. After identifying the appropriate circles, the traditional Speller then removes rest of the circles. This goes on until one circle remains. Then, the letters of this circle is put in separate circles and again the above steps are repeated until the intended letter is identified. The delete,space and stop functions delete the last character, add space and stop the program respectively. Figure 1 shows how the traditional Speller works.

WORKING OF PROPOSED SYSTEM

As mentioned earlier, our aim is to further the previous work by adding word completion and prediction feature. To achieve this goal, we use two repositories, one containing a list of 20000 words [3] which are used most often around the world and another containing 5-gram [2] widely used phrases or sentences. Both the lists are sorted in the descending order of the priorities assigned to each word/5-grams.

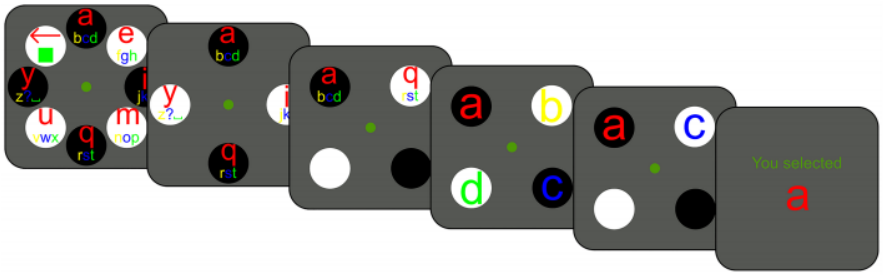

Figure 1. Letter selection in Traditional Speller

We also expect to improve the user experience gradually by updating these word list once the user starts to use the system.

Word Completion

We use the traditional Speller for retrieving the user's intended letter. Then we parse through the word list to find appropriate top 3 suggestions based on this letter. Each of the word suggestion is shown on the screen sequentially with a delay of 4 seconds as shown in figure 2. If no or less than 3 suggestions are available, none of these suggestions will be shown. The user can select any of the suggestion shown on the screen by doing an intentional eye blink. The system sees it as a blink only when it loses track of the pupil. For the intentional blink to be recognized by the system, it should be a little prolonged blink (>100ms) than just a usual eye blink. Whenever a word is chosen through such a blink, that word is added to the sentence being written along with a 'space' appended in the end. After a word is chosen, we increase the respective priority of this word in the word list. Also, if the system comes across a new or unknown word in the sentence being written, that word is added to word list. These activities are done to make sure that our upgraded system gives better or faster suggestions next time that are more suitable to that user.

Figure 2. Word suggestion for letter 'm'

Word Prediction

The system always keeps track whether a word is completed or not by checking the last character of the current sentence. If it is a 'space', it considers that a word has been written and tries to look in the 5-gram list using the previous words present in the sentence. For example, if the sentence has two words, first word will be looked in the 1st word column of the 5-gram, second word will be looked in the 2nd word column and then most suitable top 3 suggestions will be retrieved from 3rd word column and presented it to the user. Similarly, when there are more words in the sentence, all these previous words (maximum 4) will be used to give suggestions for next word. If no or less than 3 suggestions are available, none of these suggestions will be shown. The selection of these words also works the same as Word Completion. Figure 3 shows how the screen looks like.

Figure 3. Word predicted using 5-grams

If the user does not want to choose any of the suggestions, he/she should not do an intentional blink when the suggestions are shown which will ultimately take him/her back to traditional Speller to spell the next letter.

USER STUDY

In order to test our upgraded speller and to compare it with traditional Speller, we conducted a study on participants. These users were not compensated for participating. Our study explores how comfortable they were when using our system. We expect our system to be a large timer saver because of keeping the traditional Speller to minimum use.

Experimental Design and Task

The participants were instructed to write a short sentence "my name is lisa" using our upgraded Speller. The idea of this task was to compare the time taken in our upgraded Speller with the time it takes in traditional Speller. To obtain the required results, time stamps after every letter selection and task completion were recorded for each user and they were

stored as user's log information. The number of errors made while selecting letters were also recorded for each user.

As the sentence had 17 characters it would take 17 steps in traditional Speller whereas it would be around 8 steps in our upgraded Speller. Hence, we expected an efficiency improvement of approximately 50%.

Participants

Seven users, all being students from University of Groningen, were initially considered for this study. But later on, data of three participants had to be discarded as they were not able to finish the task and became frustrated.

Experimental Setup

A sophisticated eye tracker device (Model: EyeLink 1000 Plus) [4] was used to accomplish the task. This device was connected to the computer through a software which is available along with the device itself. At first, the eye tracker was calibrated to track pupil movements of each user. The experiment was run on Opensesame [5] environment using Python language. Major Python libraries used were PyLink [6], PyGaze, Psychopy, Numpy and Pandas.

Procedure

Before starting directly with the task, a small training session for participants was conducted where they were explained about the upgraded Speller and how it is to be used. To give an idea how it works, they were also allowed to run this Speller and write any sentences or words according to their choice.

All the participants started the task with full enthusiasm. But after a short while they started to struggle due to the problem of fixation: Where the user should not move their eyes away from the center and attend covertly to the circle containing the intended letter. Because of this, first three participants kept on selecting wrong letters and got frustrated. Ultimately, they asked to abort our experiment as they felt it was too tedious and taking much time of their busy schedules.

As a result, we discarded the data of these users and also removed the fixation rule from our experiment to continue with rest of the participants.

The user data of the rest four participants were recorded in separate files. We also took oral feedback from them and noted down to further improve our system.

Calculation and Measurement

In the study we measured the following:

- Calculating the total time it would take to complete the task in traditional Speller: This is done by obtaining the time taken only to spell letters by fetching respective log data of each user. Then we find out the average time it takes to spell one letter for each user. This value is multiplied by 17 to get the 'ideal time' taken by user to finish the task without making errors in traditional Speller.

  But, to calculate the 'actual time' it would take, we have to add time taken for errors too. So we calculate the time taken for errors by multiplying the number of errors with average time for one letter and again with 2. We do this because, to rectify an error the user has to select a backspace symbol and type the correct letter which adds 2 more steps to complete the task. Finally this value is added to 'ideal time' to obtain the actual time taken by the user in traditional Speller. The formula can be stated as follows:

$$\text{Time taken in traditional Speller} = \text{Avg time for one letter}*17 + \text{ Avg time for one letter}*\text{Number of Errors}*2$$

- Total time taken to complete the task using upgraded Speller: The program itself records the start and end time of the task and calculates the total time taken for each user which is then stored in user's log.
- Average time taken in traditional Speller and upgraded Speller were calculated by adding these respective values of all users and dividing them by total number of participants (i.e. 4).
- Finally, overall improvement was calculated by obtaining the difference between these two time values, dividing by average time taken in traditional Speller and then converting it to percentage. Formula is as shown below:

$$\text{Overall improvement (\%)} = \frac{\text{Avg time for traditional Speller} - \text{ Avg time for upgraded Speller}}{\text{Avg for traditional Speller}} * 100\%$$

RESULTS

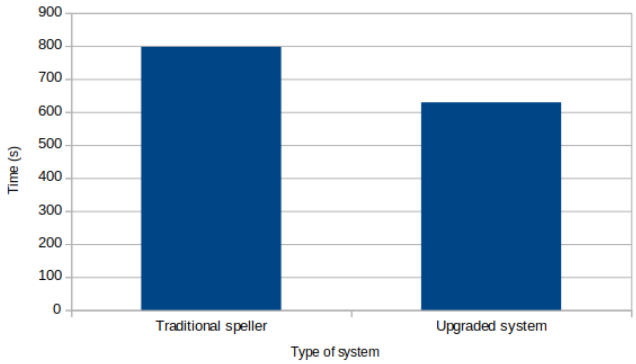

Figure 4. Bar chart of the results obtained

The results obtained were only from the study of four participants. Due to problems stated before in this paper, the data of three other participants, who did not finish the task, had to be discarded. Hence, the results shown here can be considered as biased in a way.

The average time taken for traditional Speller was estimated to be 797.4 seconds whereas time taken in our upgraded Speller was 629.2 seconds. Figure 4 shows a bar chart for the same. The overall efficiency improvement was calculated to be around 21%.

DISCUSSION

First of all, participants had trouble using the traditional speller because of which three of them got frustrated and left. We then had to drop the fixation to perform the experiment with rest of the four participants. Even after doing this, we still observed an average error rate of around 10%. The participants complained that error handling needed improvement.

Secondly, the participants had to concentrate on the screen all the time and even if they moved their head a little bit, the eye tracker lost track of the pupil and considered it as an intentional blink. Again, a re-calibration had to be done in such cases and due to this, the task completion was delayed.

Also, the fetching of suggestions took more time than expected as the program had to search through a big amount of data for every suggestion. Apart from this, we only used 5-gram suggestions for word prediction. So even for sentences only having one word, the second word was predicted from the same 5-gram list which may not always provide a better suggestion.

Lastly, the upgraded Speller cannot be easily used by anyone. The participants had to undergo a small training session in order to understand how to use it before starting with the task. Even after doing this, we still observed huge differences in individual performance.

CONCLUSION AND FUTURE WORK

One major plus point is our system is that we were able to reliably distinguish between an intentional blink and a normal blink. Blink detection proved to be a faster way for word selection. Although, looking away from the screen was also classified as an intentional blink.

Hence, keeping in mind that our upgraded Speller led to faster task completion, we can say improvements were there but not as high as expected.

One major drawback was found to be the error handling technique in our system. To eliminate this our future work will be on mainly improving the delete function which now deletes only one previous letter at a time. We also wish to include a verification step in the system which will ask the user to confirm the letter/word chosen before appending it to the sentence.

In the current system, we increase the priorities of the chosen words and add new words with priorities with fixed values. In future, we seek to find optimal parameter values to update these priorities in the word repository. We also seek to incorporate more N-grams in future to give more appropriate suggestions to the users. The amount of time these suggestions are shown on the screen will also be optimized in future.

We wish to prevent wrongful classification of blinks when user is looking away. But we are yet to come up with any sort of idea in future to resolve this issue.

To conclude, this paper explored the use of our upgraded Speller and compared it with the already existing traditional Speller. We explored both strengths and weaknesses of our system and how we seek to rectify the drawbacks in future.

REFERENCES

1. Pupillary Response
   https://en.wikipedia.org/wiki/Pupillary_response

2. N-grams Data
   https://www.ngrams.info/download_coca.asp

3. Word List **https://github.com/first20hours/google-10000-english**

4. Eye Tracker **https://www.sr-research.com/products/eyelink-1000-plus/**

5. Opensesame **https://osdoc.cogsci.nl/**

6. PyLink **https://pypi.org/project/PyLink/**

7. Sebastiaan Matht, Jean-Baptiste Melmi1, Lotje van der Linden, Stefan Van der, Stigchel: The Mind-Writing Pupil - A Human-Computer Interface Based on Decoding of Covert Attention through Pupillometry.
   https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0148805
   https://github.com/smathot/mind-writing-pupil